마침내, 붕괴되었습니다

Woojin Lee

2022-09-16

Abstract

증명은 수학에 있어 필수적이고, 나는 이 증명이 수학을 완벽하게 만든다고 믿어왔다. 힐베르트를 필두로 한 형식주의자들은 엄격한 규칙들로 이루어진 논리적 기호 집합을 통해 수학적 명제들을 기술할 수 있는 증명 가능한 형식 체계를 원했다. 하지만, 이들이 만든 수학적 체계에도 "기초 연산이 가능한 모든 수학에는 항상 증명할 수 없는 참된 명제가 있다", "유한한 시간 내에 질문에 대답할 수 있는 알고리즘이 없다"는 결점이 있음이 밝혀졌고, 이들에 대한 증명은 자가당착을 통해 이루어진다.

고사성어 지피지기 백전불태(知彼知己 百戰不殆), 소크라테스의 "너 자신을 알라(Know thyself)". 이외에도 자기 자신을 돌아보라는 뜻을 가진 문장은 수없이 많다. 아무리 성능이 좋은 카메라라 하더라도 카메라 자체는 사진에 담을 수 없다. 당연하지만 곱씹어 보면 신기한 이러한 사실을 수학도 피해갈 수 없다. 힐베르트(David Hilbert)는 1과 1의 합이 2라는 수준의 직관조차 배제하기 위해 최대한 엄밀한 수학의 체계를 만들고자 노력하였다. 애매모호한 부분이 많은 인간의 말을 지양하고, 엄격한 규칙들로 이루어진 논리적 기호 집합을 통해 수학적 명제들을 기술할 수 있는 증명 가능한 형식 체계를 원한 것이다. 슬프지만, 심혈을 기울여 만든 힐베르트의 형식 체계에도 완전성과 결정 가능성이라는 구멍이 존재한다. 필자는 수학에는 탄탄한 증명이 뒷받침되기에 애매모호함이 절대 존재할 수 없다고 믿어왔다. 하지만, 30분 남짓의 영상을 본 후에는 영화 "헤어질 결심"의 유명한 대사처럼, '마침내', '붕괴'했다.

힐베르트의 형식 체계는 **완전(Complete)**하지 않다. 완전하다는 것은, 참인 모든 문장들을 증명할 수 있는 방법이 항상 존재한다는 것을 의미한다. 학부 2학년 과목 이산구조에서 다룬 논리연산은 완전하다. 하지만, 고차 논리로 확장하면 논리 체계는 완전하지 않게 된다. 괴델(Kurt Gödel)은 기초 연산이 가능한 그 어떠한 수학 체계에서도 증명할 수 없는 참된 명제가 항상 존재할 것이다는 것을 보였다. 참인 것을 알지만 증명 불가능하다니. "괴델 수 g는 증명할 수 없다"는 명제에 대한 증명은 "괴델 수 g는 증명할 수 없다" 그 자체이다. g가 거짓이라면, 일관성(Consistency)에 흠이 간다. g가 참이라면, 증명할 수 없는 참인 명제가 존재함으로써 완전하지 않은 체계가 된다. 모순이다. 엄밀성을 부여하기 위해 증명을 해 왔는데, 증명할 수 없는 명제가 결국에는 존재한다는 것이 참으로 신비하다.

영상에서는 **결정 가능성(Undecidability)**에 대해서도 다룬다. 결정 가능하다는 것은, 어떠한 명제가 공리를 따르는지 여부를 결정할 수 있는 알고리즘이 존재한다는 것을 의미한다. 앨런 튜링(Alan Turing)의 "정지 문제(Halting Problem)"가 이에 대한 실마리를 제공한다. 1936년, 튜링은 정지 문제를 해결할 수 있는 일반화된 방법이 없음을 보였다. 이 또한 위의 완전성 문제처럼 자기 모순을 통해 보일 수 있다. 어떠한 문제가 결정 불가능한지를 보이려면, 정지 문제로 바꾸어 생각해 볼 수 있는지를 보이면 된다. 조금 더 확장하면 라이스의 정리인데, 이는 튜링 머신(Turing Machine)이 받아들이는 내용이 자명(trivial)한가, 자명하지 않은가 결정하는 것은 불가능하다는 것이다. 수학이 결정 가능하지 않다는 것은, 쌍둥이 소수 추측(Twin prime conjecture)에 대한 증명이 "참 혹은 거짓"이라는 이분법적 결론이 아닌, "알 수 없음"이라는 새로운 방향 또한 생각해 봐야 한다는 것을 암시한다.

지금까지 공부하면서 적지 않은 프로그램을 작성하고, 수학 문제들을 풀어왔지만, 자체의 본질에 대해서는 생각해 본 기회가 많지 않았다. 영상에서 전하고자 하는 내용을 얼마나 깊게 이해했는지는 모르겠지만, 엄밀하다고 믿어온 수학에도 흠이 있다는 것, 그 자체만으로 흥미로웠다. 영상에서 주요하게 설명한 완전성과 결정 가능성 문제는 결국 자가당착에 직면했다. 결점을 보이기 위해 또다른 도구를 들고 오는 것이 아니라 그 자체의 모순을 보이는 과정이 깔끔하고 아름답게 느껴졌다.

힐베르트는 엄격하면서 증명 가능한 수학적 체계를 원했지만, 완벽하지 않았다. 하지만, 그의 결정 가능성에 대한 질문은 튜링 머신을, 튜링 머신은 현대 컴퓨터를 있게 했다. 교내 커뮤니티에 한 교수님이 올린 "전산학은 계산(컴퓨팅)에 관련된 모든 것에 관한 학문입니다"라는 제목의 글이 생각났다. 계산을 단순한 사칙연산이 아니라, 현실 세계의 여러 가지 것들을 **추상화**하고 **자동화**하는 것이라 설명했다. 전산학에 관련한 일을 한다면,

원하던 원하지 않던 추상화와 자동화에 대한 고민을 하게 될 것이다. 또한, 대학원을 생각하고 있는 나는 이러한 고민의 과정에서 "내가 정의한 문제가 필요하긴 한 걸까"라는 회의감에 빠지는 일이 부지기수일 것이다. 부정적인 생각이 들 때마다 역설이 현대 컴퓨터의 기반이 된 것처럼 의미 없는 물음은 없다는 자신감을 얻을 수 있지 않을까 하는 생각과 함께 글을 마친다.